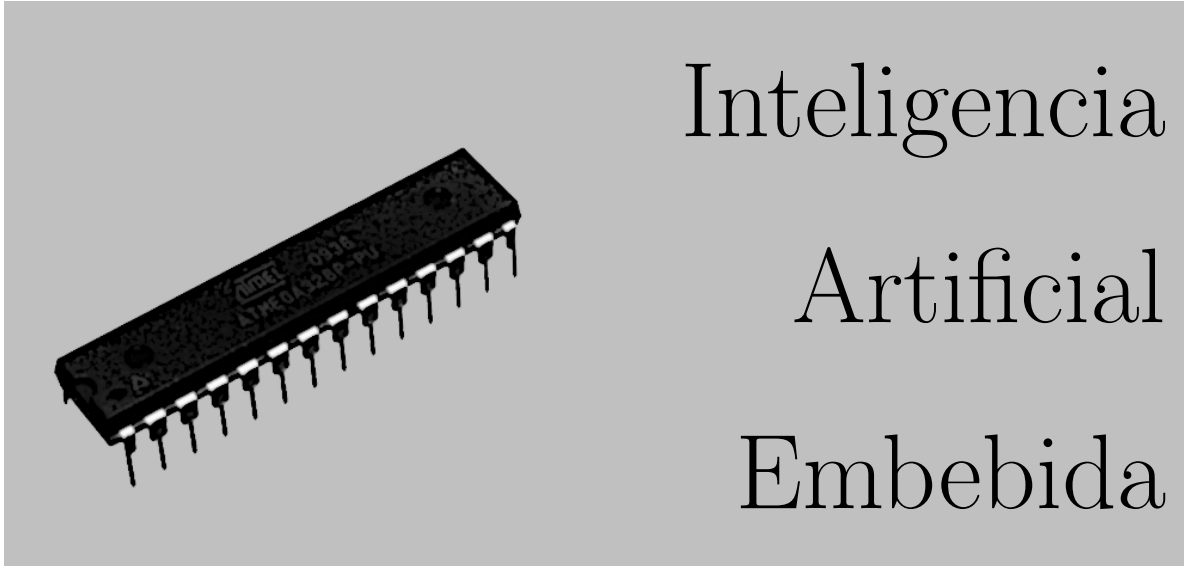




Inteligencia Artificial Embebida

Nombre: _____ Grupo: _____

Dr. Enrique García Trinidad
Tecnológico de Estudios Superiores de Huixquilucan
<https://enriquegarcia.xyz>
enrique.g.t@huixquilucan.tecnm.mx

Práctica 12: Clasificación con Árboles de Decisión

1. Introducción Teórica

Un **Árbol de Decisión** es un modelo de aprendizaje supervisado no paramétrico utilizado tanto para clasificación como para regresión. El objetivo es crear un modelo que prediga el valor de una variable objetivo aprendiendo reglas de decisión simples inferidas de las características de los datos. En esta práctica, los estudiantes construirán un árbol de decisión para predecir si se jugará al tenis en función del clima, y visualizarán la estructura del árbol generado.

Para dividir los nodos del árbol, el algoritmo necesita una métrica de impureza o ganancia de información. En este caso, utilizaremos la **entropía**. Un paso crucial en este modelo es el preprocesamiento: los árboles de decisión en **scikit-learn** requieren datos numéricos, por lo que convertiremos la información categórica (como Soleado o Lluvioso) en información numérica antes de entrenar.

2. Desarrollo de la Práctica e Implementación

2.1. Paso 1: Importar dependencias

Importamos **pandas** para el manejo del conjunto de datos, **numpy** para el manejo de arreglos, **tree** de **sklearn** para instanciar el modelo, y **pydotplus** para generar el diagrama visual del árbol.

```
import pandas as pd
import numpy as np
from sklearn import tree
import pydotplus
```

2.2. Paso 2: Función para generar el árbol de decisión

Definimos la función **createTree**. Esta separa la matriz de características (todas las columnas excepto la última) y las etiquetas (la última columna). Luego, instancia un clasificador de árbol de decisión utilizando la entropía como criterio de división (**criterion='entropy'**) y entrena el modelo mediante el método **fit**.

```
# Generar un arbol de decision
def createTree(trainingData):
    data = trainingData.iloc[:, :-1] # Matriz de características
    labels = trainingData.iloc[:, -1] # Etiquetas
    trainedTree = tree.DecisionTreeClassifier(criterion="entropy") #
        Clasificador
    trainedTree.fit(data, labels) # Entrenar el modelo
    return trainedTree
```

2.3. Paso 3: Visualización del diagrama del árbol

Esta función exporta la estructura del árbol entrenado utilizando el formato Graphviz y lo guarda como un archivo PDF en la máquina local mediante pydotplus.

```
def showtree2pdf(trainedTree, filename):
    dot_data = tree.export_graphviz(trainedTree, out_file=None) #
        Exportar formato Graphviz
    graph = pydotplus.graph_from_dot_data(dot_data)
    graph.write_pdf(filename) # Guardar el diagrama como PDF
```

2.4. Paso 4: Función para generar datos vectorizados

En la función data2vector, la instrucción `pd.Categorical(data[i]).codes` obtiene una lista de números de secuencia correspondientes a los datos originales, permitiendo convertir la información categórica (texto) en información numérica comprensible por el algoritmo.

```
def data2vector(data):
    names = data.columns[:-1]
    for i in names:
        col = pd.Categorical(data[i])
        data[i] = col.codes
    return data
```

2.5. Paso 5: Invocar las funciones y predecir

Finalmente, leemos los datos de entrenamiento desde el archivo `tennis.txt`, los vectorizamos, creamos el árbol de decisión y exportamos su visualización a `tennis.pdf`. Una vez entrenado, predecimos una nueva muestra con valores vectorizados `[0, 0, 1, 1]` (Clima soleado, temperatura baja, humedad alta, viento fuerte).

```
data = pd.read_table("./ML/tennis.txt", header=None, sep="\t") #
    Leer datos
trainingvec = data2vector(data) # Vectorizar datos
```

```
decisionTree = createTree(trainingvec) # Crear el arbol
showtree2pdf(decisionTree, "tennis.pdf")

# Predecir una nueva muestra
testVec = [0,0,1,1]
print(decisionTree.predict(np.array(testVec).reshape(1,-1))) #
    Predecir
```

3. Conclusión

El árbol de decisión logró estructurar las reglas a partir de los datos meteorológicos de entrenamiento midiendo la ganancia de información (entropía). Al proporcionarle un nuevo vector de prueba [0,0,1,1], el clasificador predice exitosamente la etiqueta ['Y'] (Sí), sugiriendo que las condiciones climáticas son aptas para jugar tenis.

4. Evaluación de Comprensión

Instrucciones: Selecciona la respuesta correcta para cada una de las siguientes preguntas.

1. ¿Cuál es la principal diferencia funcional entre un Árbol de Decisión y la Regresión Lineal en el contexto de estos laboratorios?
 - A. La Regresión Lineal se usa para clasificación de imágenes, mientras que el Árbol de Decisión predice datos numéricos.
 - B. El Árbol de Decisión requiere iteraciones manuales con la tasa de aprendizaje α .
 - C. El Árbol de Decisión aprende reglas estructuradas en forma de nodos y ramas para predecir clases o valores, mientras que la Regresión Lineal ajusta una línea recta matemática a los datos.
 - D. No hay diferencia, ambos usan obligatoriamente la función logística (sigmoide).
2. ¿Por qué fue estrictamente necesario usar la función `pd.Categorical().codes` en el Paso 4 de la práctica?
 - A. Para encriptar los datos meteorológicos por motivos de privacidad.
 - B. Para convertir las categorías de texto (ej. "Soleado", "Lluvioso") en códigos numéricos, dado que el modelo de `scikit-learn` requiere entradas matemáticas/numéricas.
 - C. Para aumentar artificialmente el número de muestras del conjunto de entrenamiento.

- D. Para normalizar los datos de modo que la media sea 0 y la varianza sea 1.
3. En la línea `DecisionTreeClassifier(criterion='entropy')`, ¿qué papel juega el parámetro `entropy`?
- A. Es la métrica o criterio utilizado por el algoritmo para medir la impureza de un nodo y decidir cuál característica es la mejor para dividir (ramificar) los datos en ese punto.
 - B. Define el tamaño de letra del diagrama PDF generado.
 - C. Es el número máximo de hojas que el árbol de decisión puede generar.
 - D. Sirve para ignorar las características que contienen valores nulos o incompletos.
4. ¿Qué biblioteca de Python se utiliza explícitamente en el código para tomar el formato de árbol generado por scikit-learn y exportarlo como un archivo gráfico en formato PDF?
- A. `matplotlib.pyplot`
 - B. `numpy`
 - C. `pandas`
 - D. `pydotplus`
5. Según el resultado de la práctica, al introducir la muestra de prueba `[0,0,1,1]` en la función `predict()`, ¿cuál fue el resultado devuelto por el árbol de decisión entrenado?
- A. Un error matemático de división por cero.
 - B. Una probabilidad decimal como 0,85.
 - C. La etiqueta de clase `['Y']` (Sí).
 - D. La etiqueta de clase `['N']` (No).